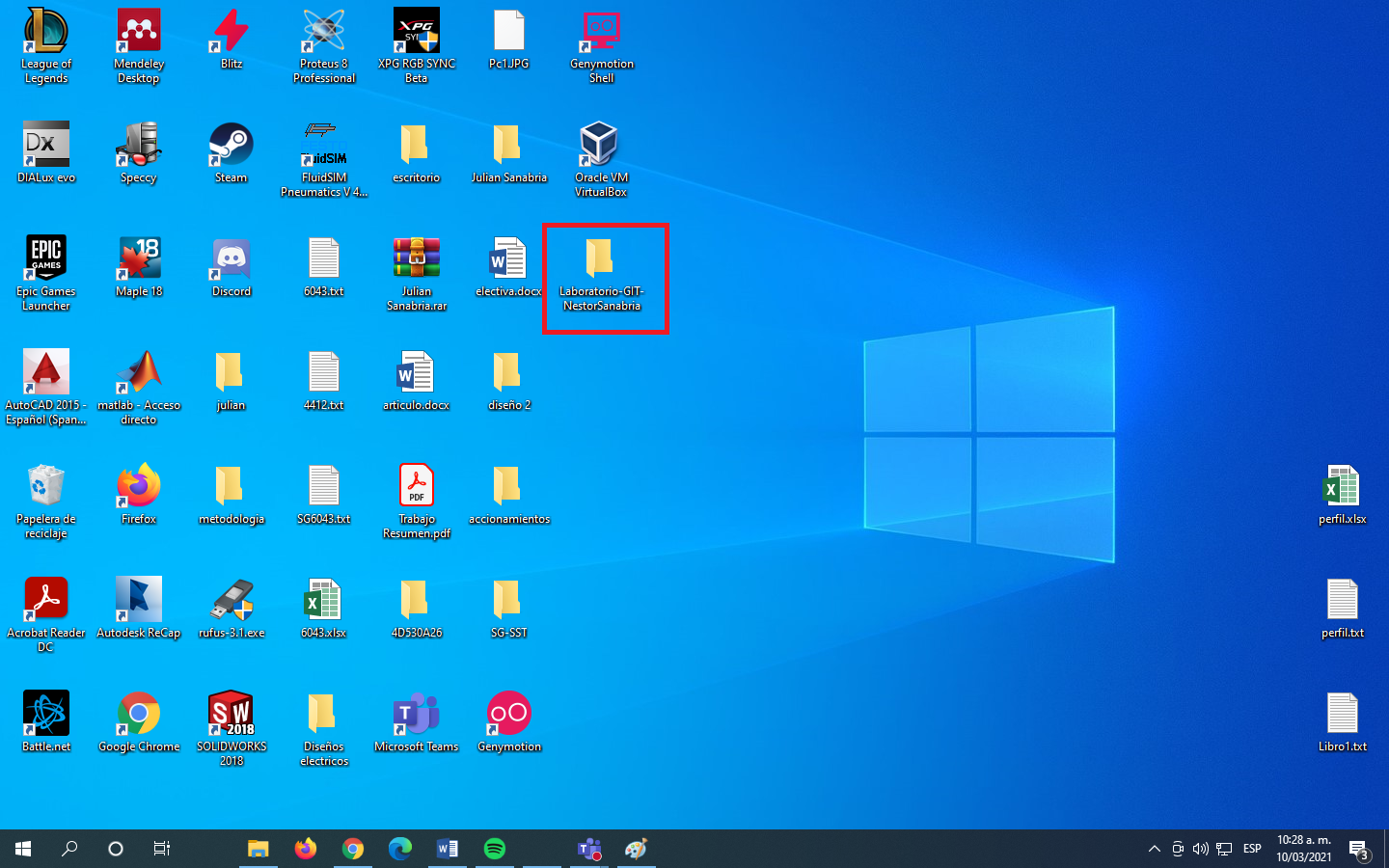
Laboratorio práctico Fundamentos de Git y GitHub

Siga las instrucciones indicadas a continuación:

# 1

Cree una carpeta en su escritorio llamada Laboratorio-GIT-SuNombreUsuario, ejemplo: Laboratorio-GIT-jahigueram

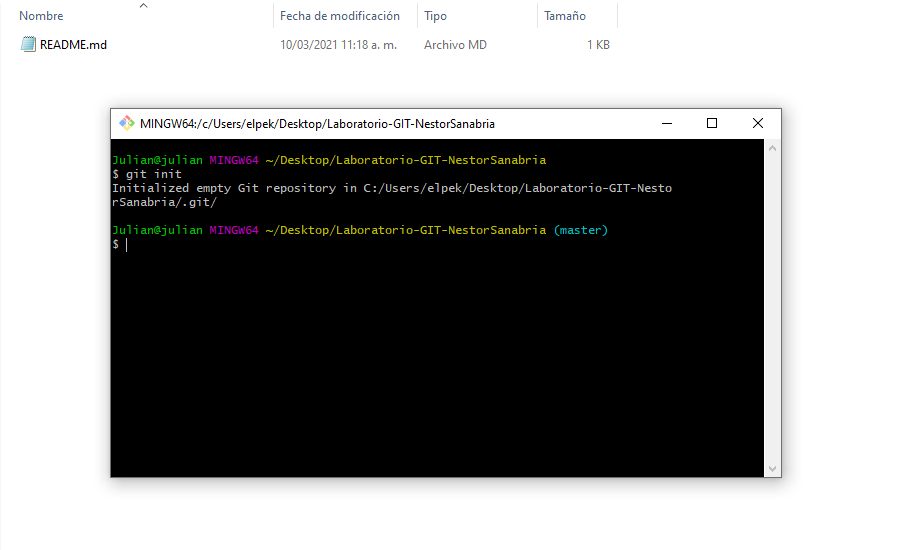
* Se crea la carpeta respectvamente con el nombre “Laboratorio-GIT-NestorSanabria”



# 2

Dentro de esa carpeta, inicialice el proyecto de git

* Como podemos ver en la imagen ejecutamos el comando git init e inicializamos o creamos el repositorio de la carpeta



# 3

Dentro del repositorio (o proyecto) cree un archivo llamado README.md, ingrese el siguiente contenido:

# Información

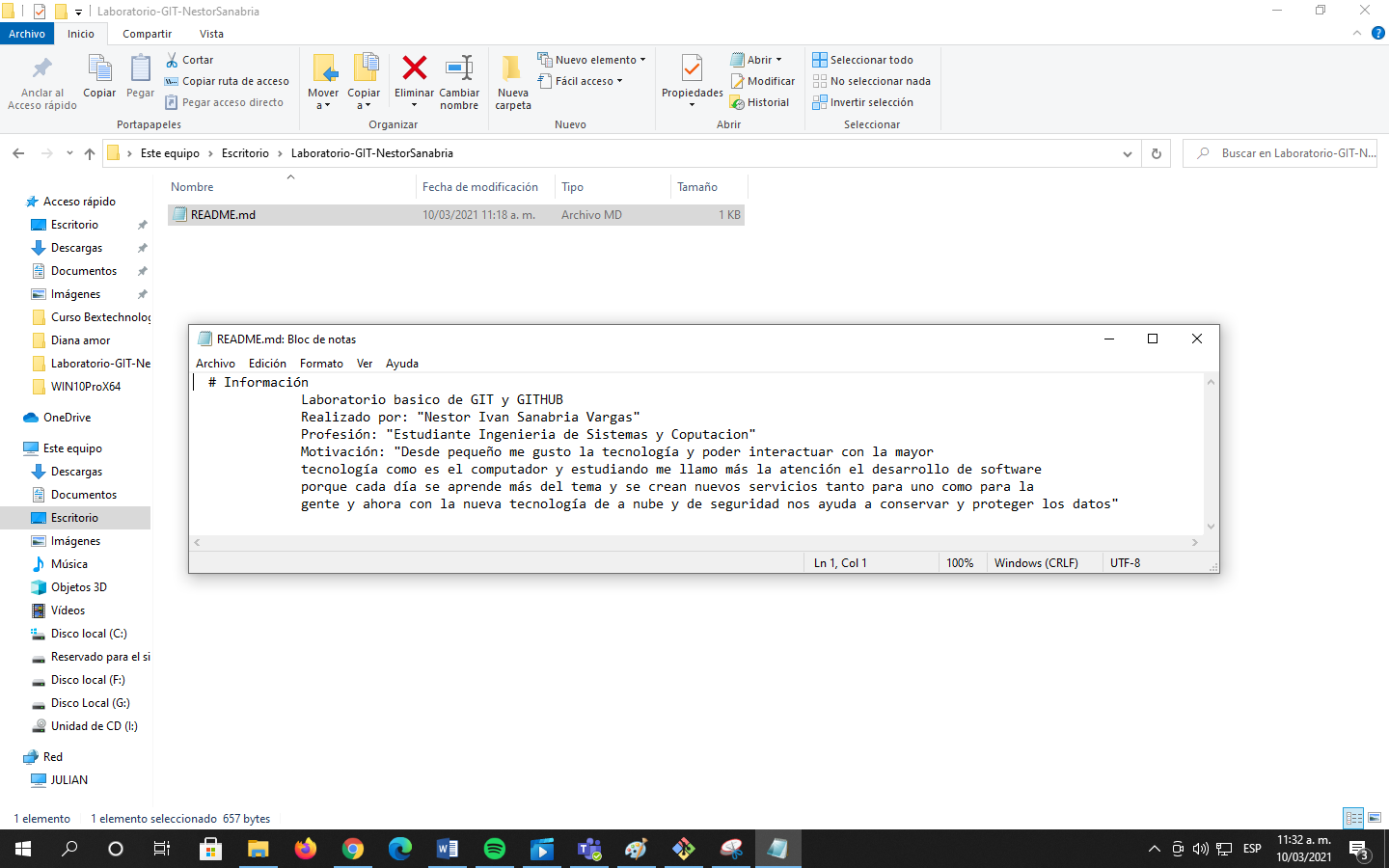
Laboratorio básico de GIT y GITHUB

Realizado por: "Nombre completo y apellidos"

Profesión: "Nombre de su profesión"

Motivación: "Breve descripción que explique tu motivación principal de pertenecer a un área de desarrollo de software"

* Como podemos ver en la imaguen se crea el archivo Readme.md con los respectivos datos solicitados

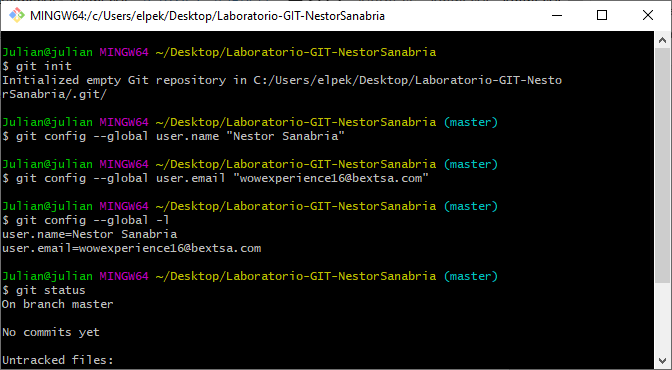


# 4

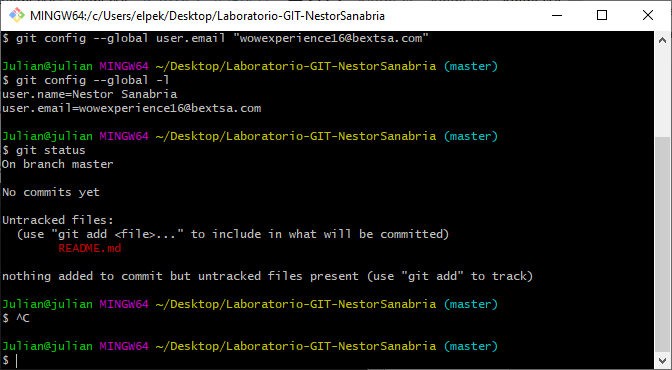
Agregue el archivo README.md al stage o escenario (Para que empiece a ser el archivo trackeado)

Realice el primer commit con el mensaje Creación del archivo readme.

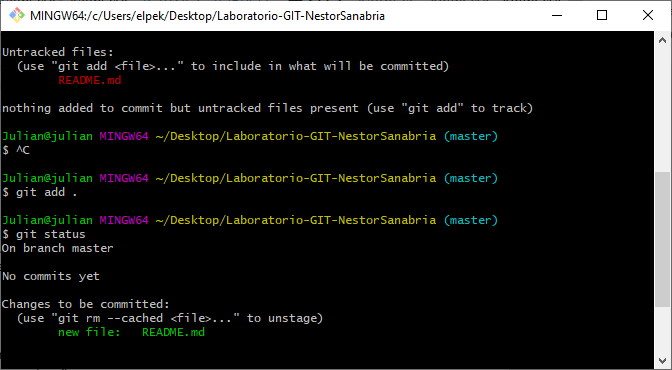
* Para subir el archivo a stage antes debemos hacer diferentes pasos como inicializar el repositorio con “git init”, despues creamos usuario con “git config --global user.name “Nestor Sanabria” y correo con “git config --global user.email “[wowexperience16@bextsa.com](mailto:wowexperience16@bextsa.com)”, a continacion ejecutamos el comando “git config –global –l” para confirmar q e usuario y el correo estén correctamente ejecutados.



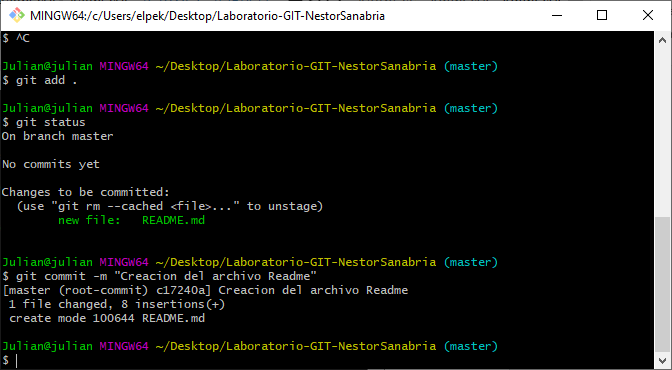
* Al hacer el “git status” nos damos cuenta que el archivo README.md está listo para subirlo al repositorio



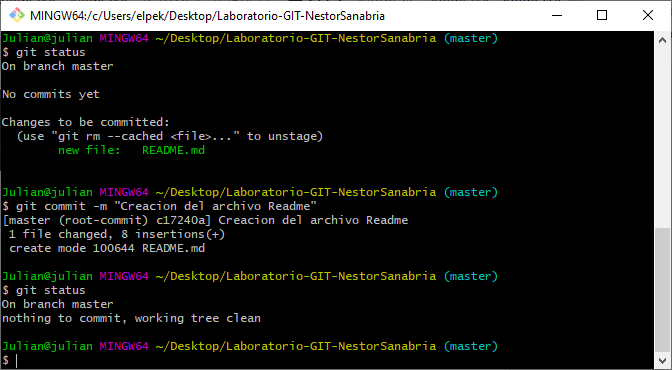
* Para tener un archivo Trakeado lo primero que se hace es ejecutar el comando “git add .” lo que hace es mover archivos de Unstaged al estado Staged q como podemos ver en a imaguen cambio el archivo Readme.md de rojo a verde.



* Lo segundo que se hace para tener un archivo Trakeado es ejecutar el comando “git commit –m “Creación del archivo readme” que nos ayuda a mover archivos de Unstaged a Tracked. Esta es una ocasión especial, los archivos han sido guardados o actualizados en el repositorio.



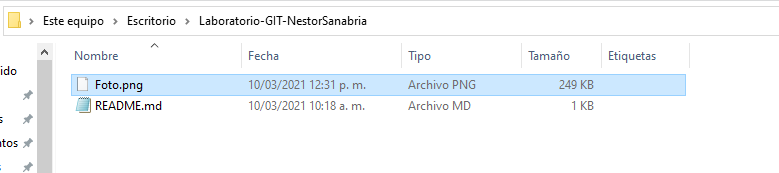
* Como podemos ver al darle “git status” no hay ningún archivo para subir lo que quiere decir que el archivo Readme ya se encuentra en el repositorio



# 5

Dentro del repositorio (o proyecto) copie una foto suya de formato de tipo hoja de vida o de cedula llamada foto.png

* Agregando foto con formato “.png”

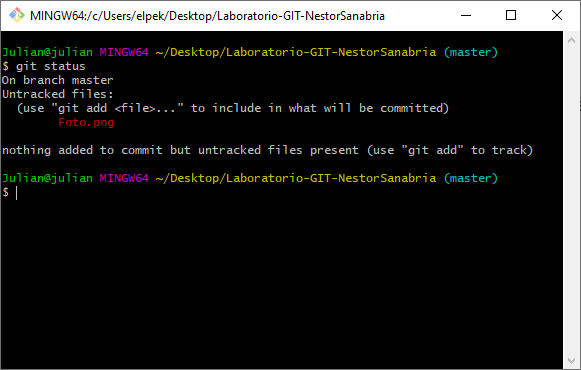


# 6

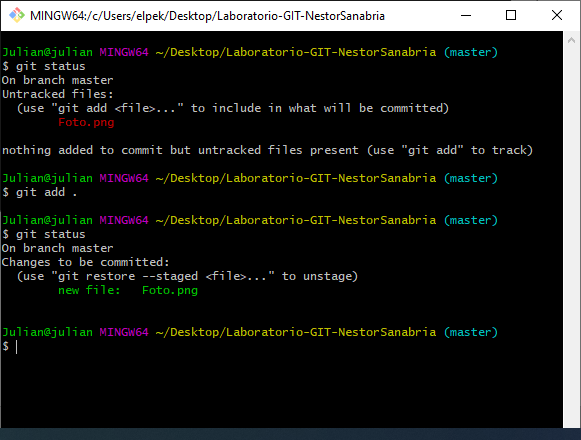
Agregue el archivo foto.png al stage o escenario (Para que empiece a ser el archivo trackeado)

Realice el segundo commit con el mensaje Subida foto de hoja de vida

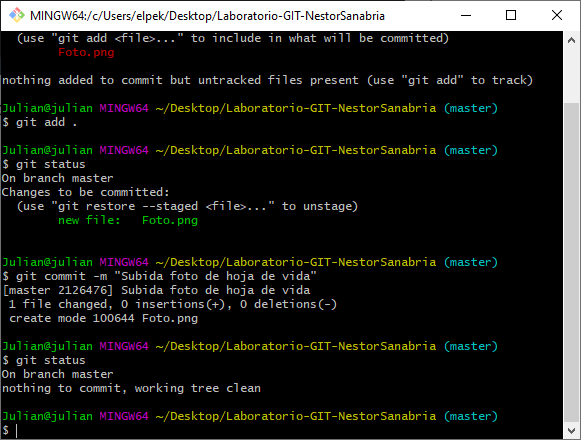
* Al hacer el “git status” nos damos cuenta que el archivo “foto.png” está listo para subirlo al repositorio



* Para tener un archivo Trakeado lo primero que se hace es ejecutar el comando “git add .” lo que hace es mover archivos de Unstaged al estado Staged q como podemos ver en a imaguen cambio el archivo Foto.png de rojo a verde.



* Lo segundo que se hace para tener un archivo Trakeado es ejecutar el comando “git commit –m “Subida foto de hoja de vida” que nos ayuda a mover archivos de Unstaged a Tracked. Esta es una ocasión especial, los archivos han sido guardados o actualizados en el repositorio. Y con el “git status” comprobamos que la foto está en el repositorio

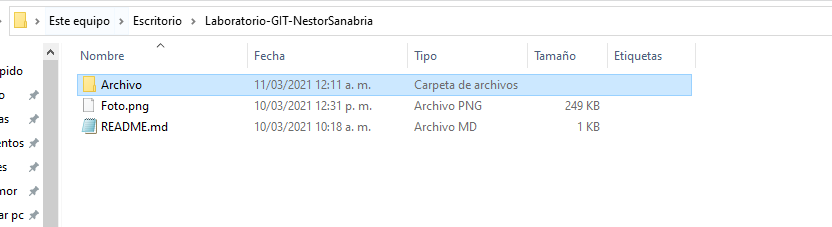


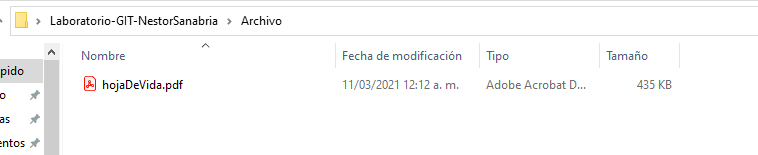
# 7

Cree una carpeta dentro del proyecto llamada Archivo

Dentro de la carpeta Archivo, copie su hoja de vida en formato .pdf hojaDeVida.pdf

* Se crea la carpeta Archivo y se copia la hoja de vida.pdf

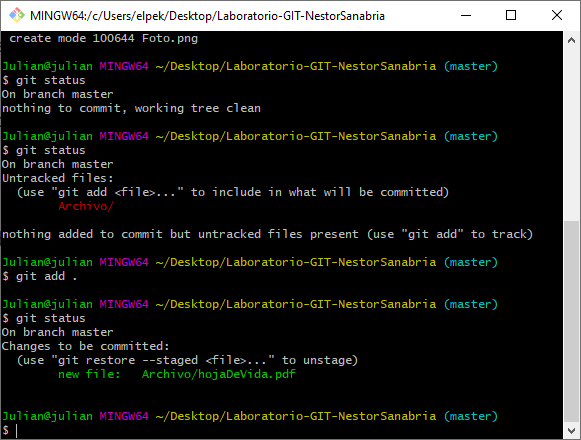




# 8

Agregue los nuevos archivos creados (El folder y el archivo) hojaDeVida.pdf al stage (Para que empiece a ser el archivo trackeado)

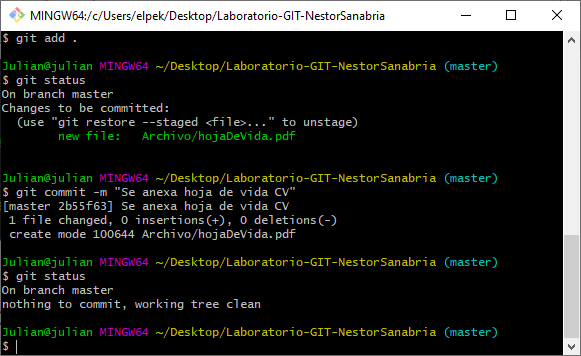
* Se sube la carpeta y la hoja de vida a repositorio con “git add .” y “git status” para saber si subió el archivo y la carpeta cando estén en color verde



# 9

Realice el tercer commit con el archivo hojaDeVida.pdf, y en el mensaje coloque Se anexa hoja de vida CV

* Se ejecuta el “git commit –m “Se anexa hoja de vida CV” para subir la carpeta y el pdf al repositorio



# 10

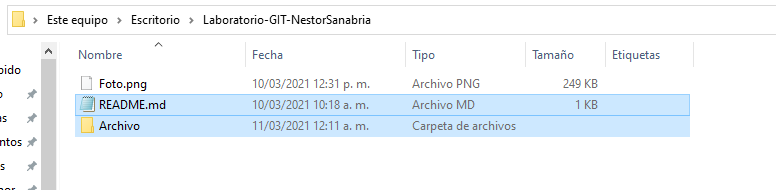
Por favor para cada uno de los pasos a continuación vaya tomando capturas de pantalla y almacene las en un lugar diferente a la carpeta del proyecto que está trabajando con GIT

Borre la carpeta Archivo

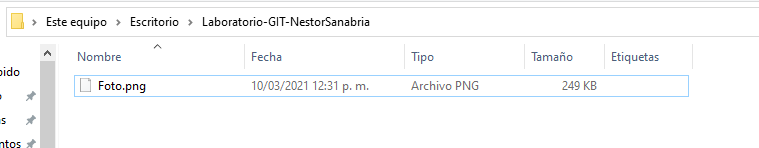
Borre el archivo README.md

El único archivo que debe de quedar, es foto.png

* Los archivos están listos para borrar



* Los archivos están borrados

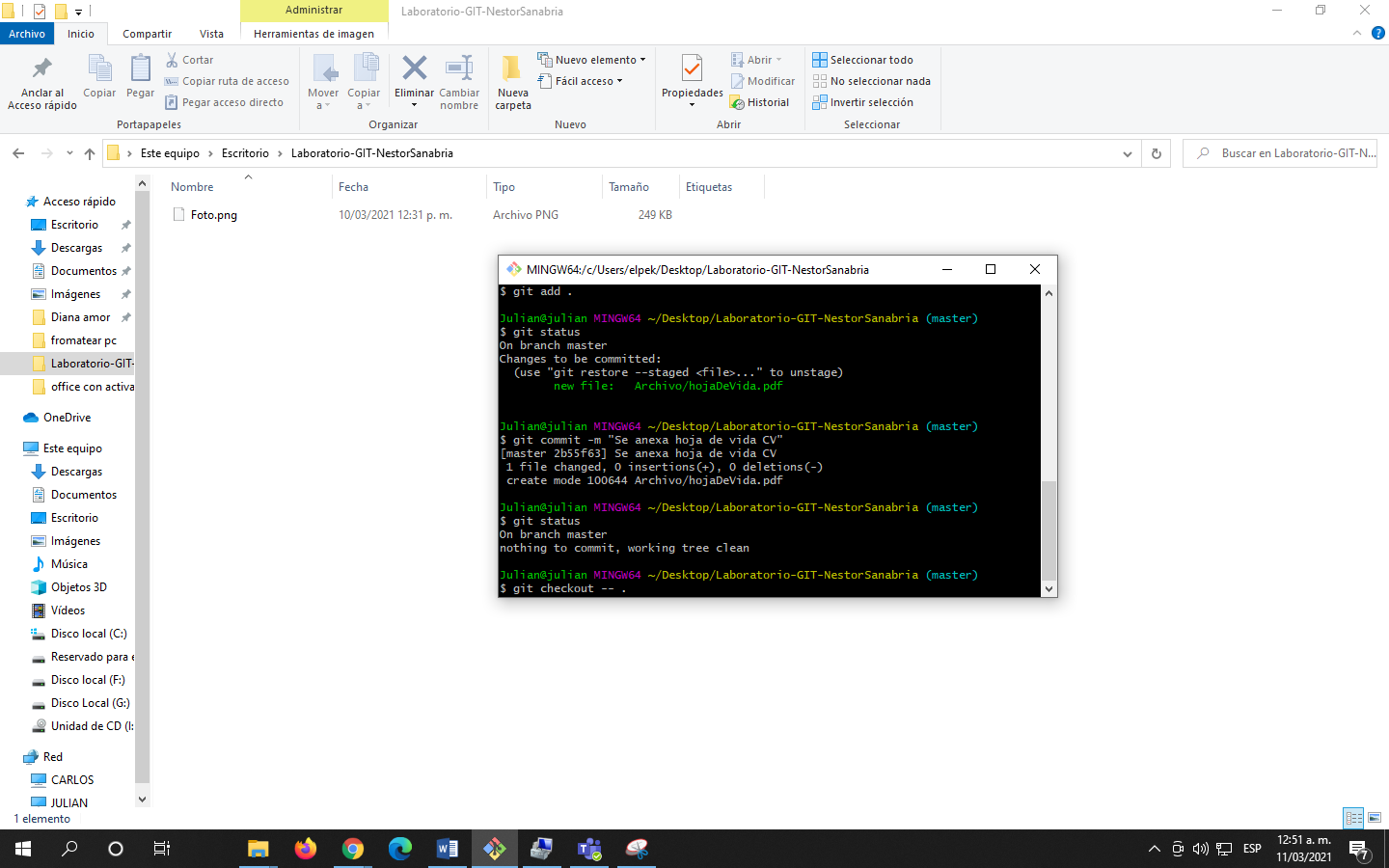


# 11

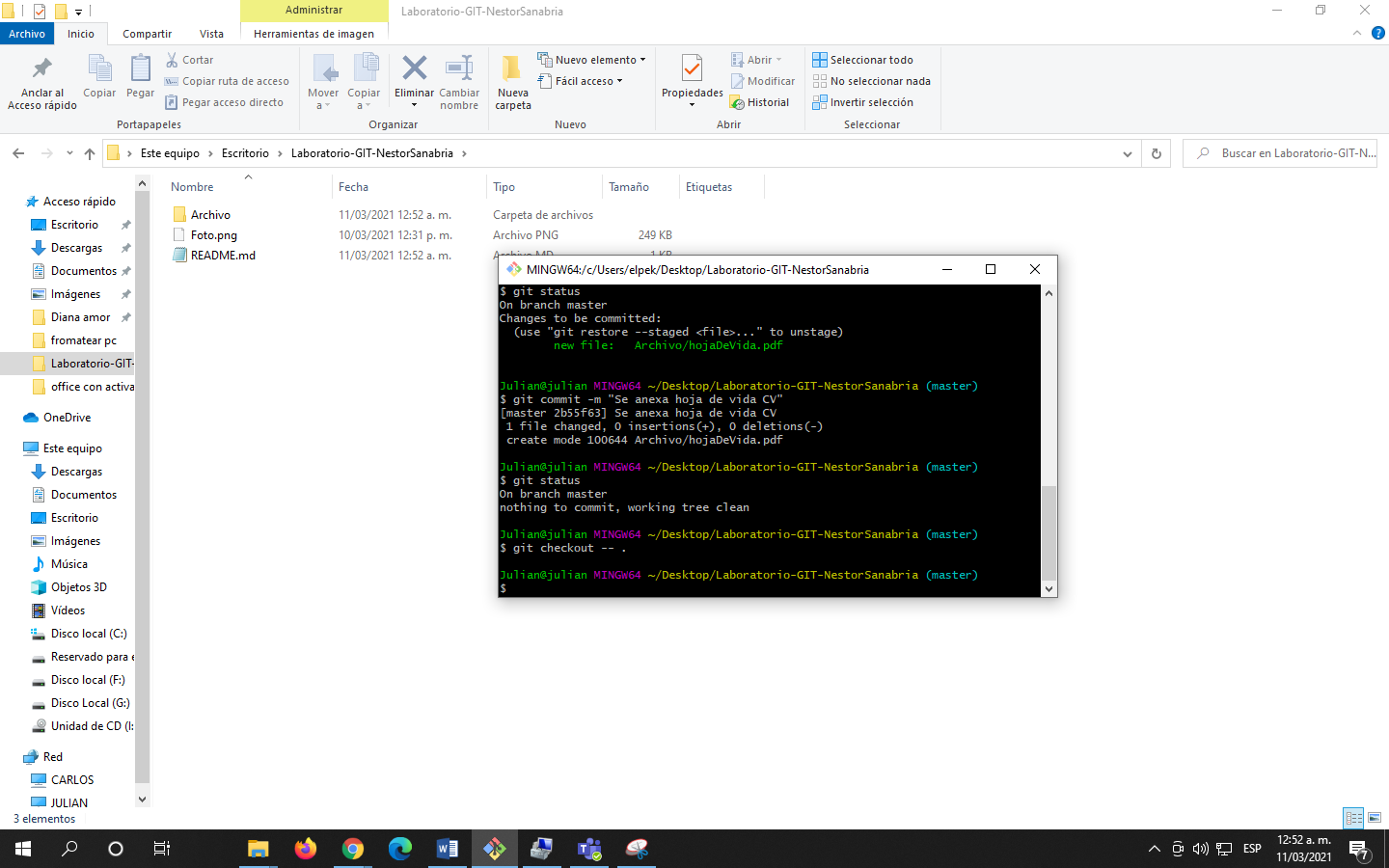
Por favor para cada uno de los pasos a continuación vaya tomando capturas de pantalla y almacénelas en un lugar diferente a la carpeta del proyecto que está trabajando con GIT

Reconstruir todo lo borrado con un único comando. (Los archivos del punto 10 deben de reaparecer)

* A continuación en la imagen se ejecuta el comando “git checkout -- .” para recuperar los elementos eliminados



* Como podemos ver aparecen los archivos borrados

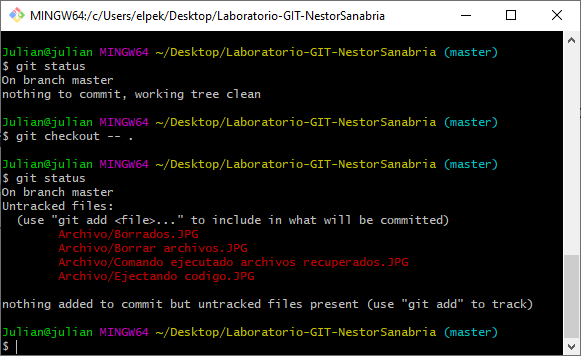


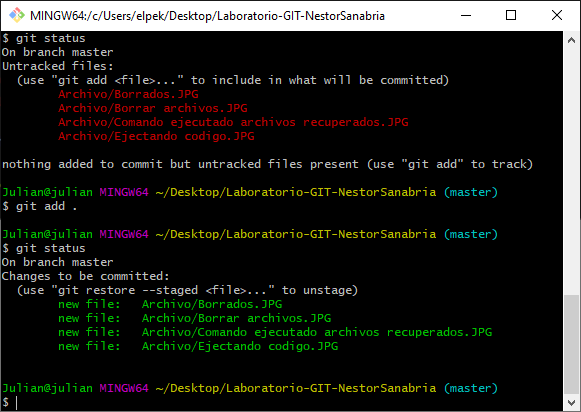
# 12

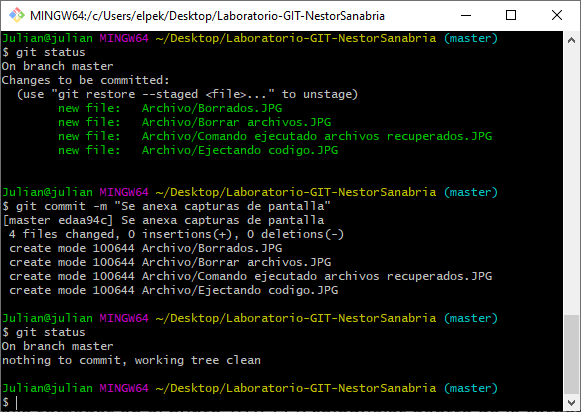
Copie las capturas de pantalla tomadas en el punto 10 y 11 y coloquelas dentro de la carpeta del proyecto Git llamada Archivo

Agregue los archivos copiados al stage (Para que empiecen a ser trackeados)

Realice el cuarto commit y en el mensaje coloque Se anexa capturas de pantalla

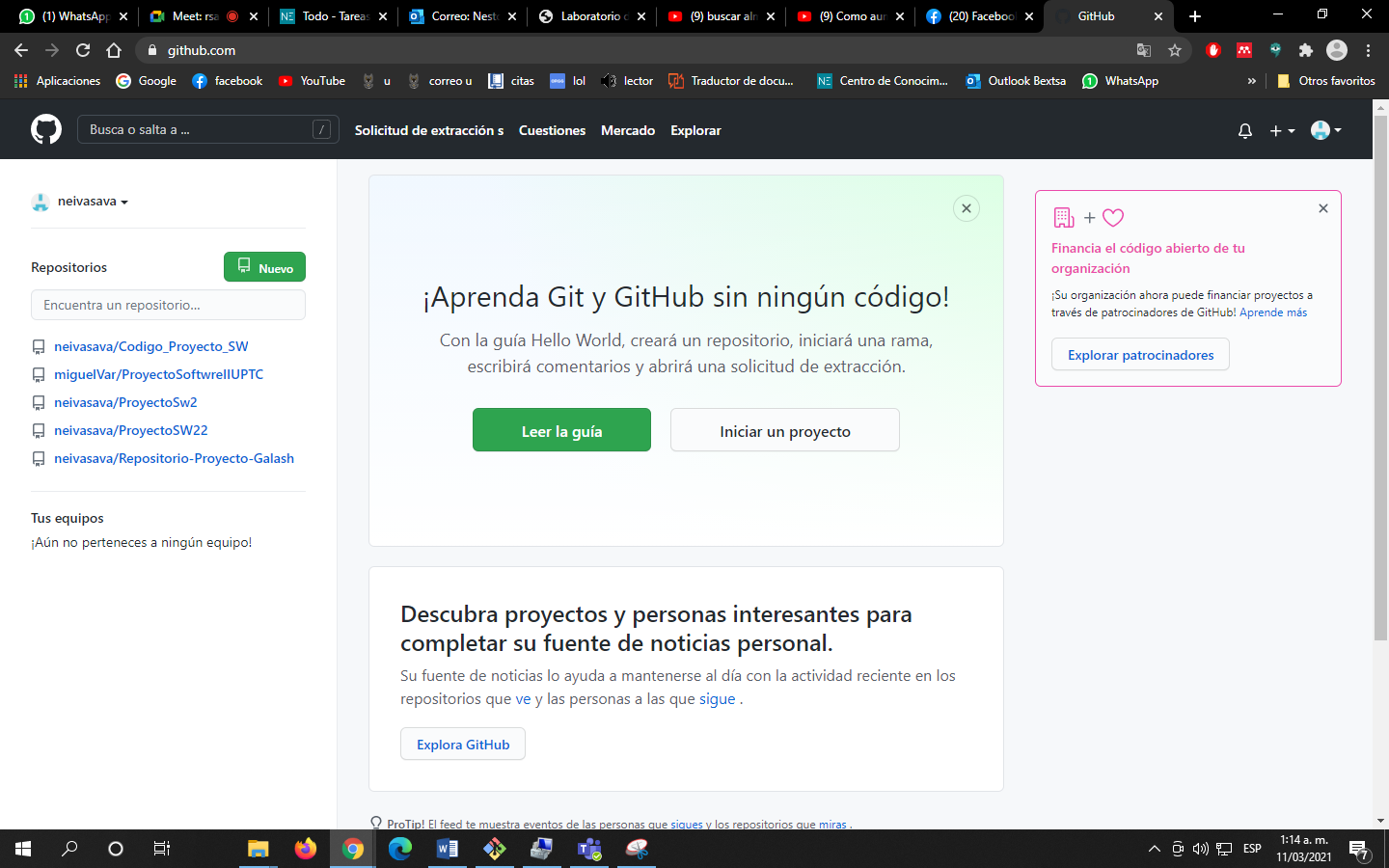






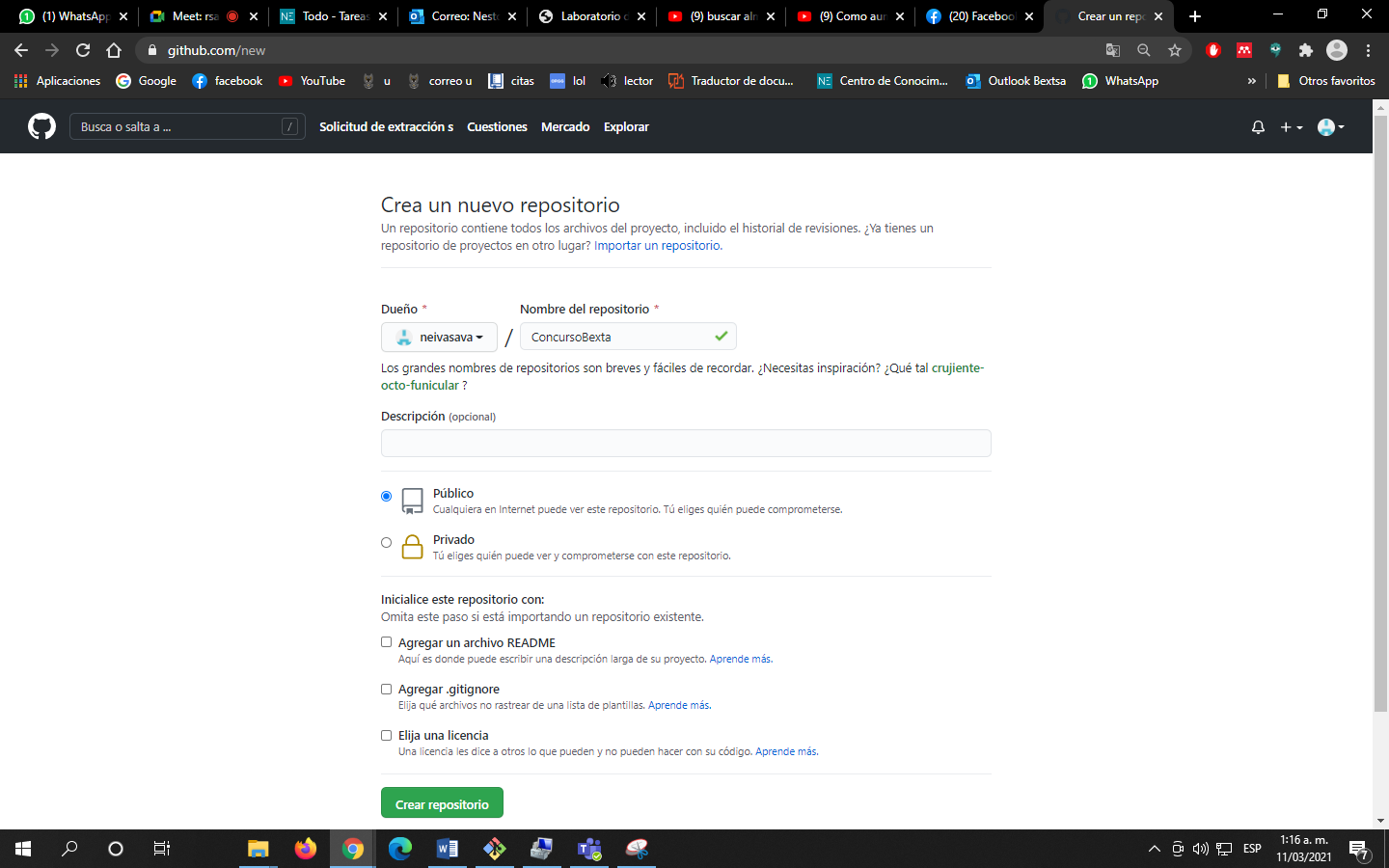
# 13

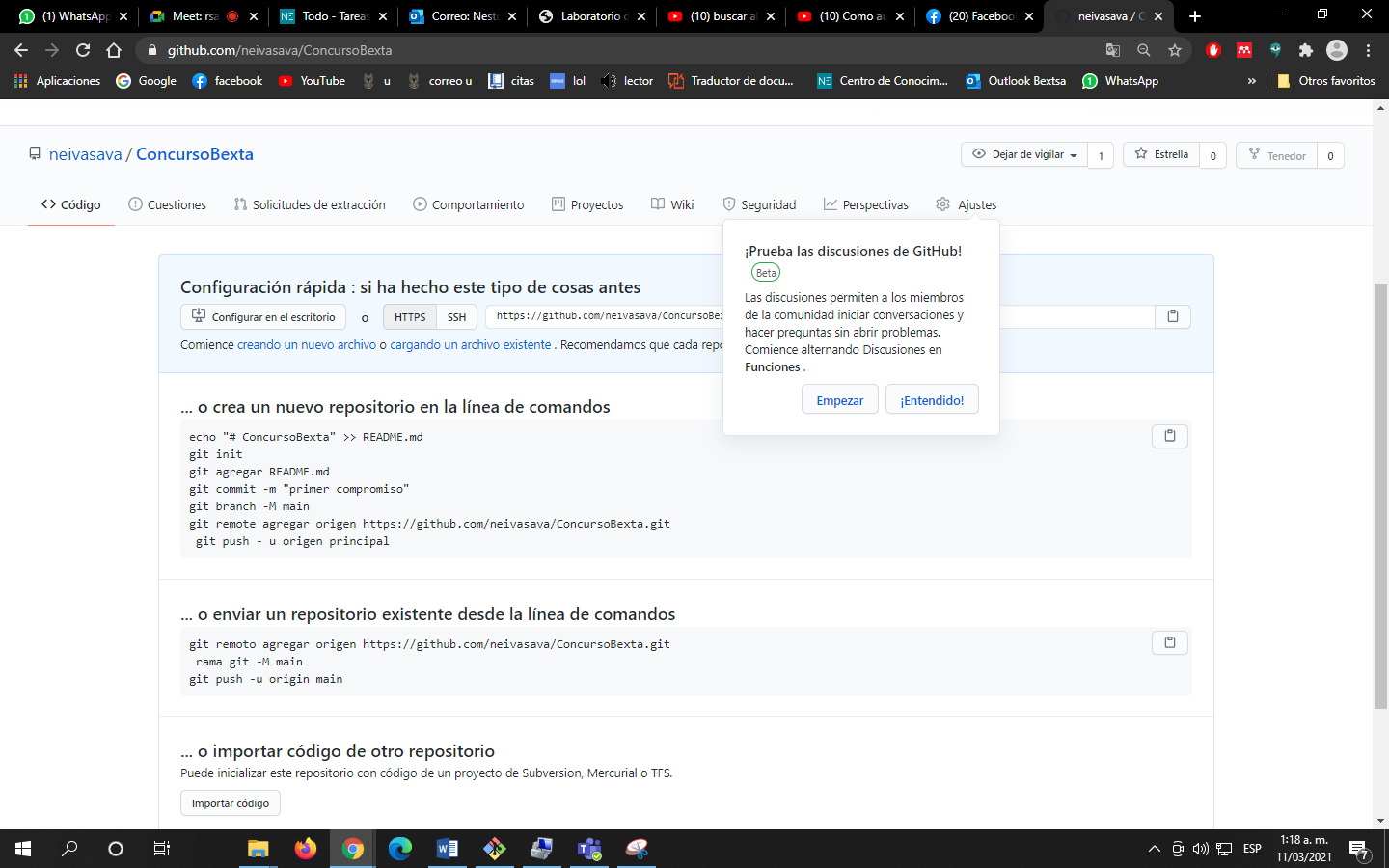
Cree una cuenta de Git HUB (Si ya la tiene creada puede utilizarla)



# 14

Cree un proyecto de tipo público llamado ConcursoBexta





# 15

Suba el proyecto (Repositorio local de Git que ha venido trabajando de manera local) al proyecto de Git Hub en la nube llamado ConcursoBexta

# 16

Finalmente, se debe enviar la URL del repositorio de Git Hub desde tu cuenta del correo del concurso a julianh@bextsa.com

Gracias por la participación